**Métodos de ordenação**

**Bubble**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Colunas1** | **Colunas2** | **Colunas3** | **Colunas4** | **Colunas5** | **Colunas6** | **Colunas7** |
| Qtd entrada | Cres. Dist | Cres. Rep | Decres. Dist | Decres. Rep | Alea. Dist | Alea. Rep |
| 120k | 4890 | 4894 | 14176 | 14017 | 19882 | 20001 |
| 180k | 10972 | 10950 | 31611 | 31525 | 45504 | 44040 |
| 240k | 19556 | 19464 | 56114 | 56030 | 80933 | 78176 |
| 300k | 30596 | 30479 | 87670 | 87695 | 126210 | 122137 |
| 360k | 44077 | 44026 | 126207 | 126362 | 182085 | 175620 |
| 420k | 60251 | 60031 | 172060 | 171981 | 264388 | 240741 |
| 480k | 78226 | 78212 | 224764 | 224889 | 357745 | 312315 |
| 540k | 99445 | 99174 | 284592 | 284526 | 414703 | 396209 |
| 600k | 123348 | 123408 | 351645 | 351934 | 508504 | 488877 |
| 660k | 150109 | 150206 | 425698 | 425905 | 622629 | 592157 |
| 720k | 179313 | 179208 | 507037 | 507076 | 769964 | 704878 |

**Insertion**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Colunas1** | **Colunas2** | **Colunas3** | **Colunas4** | **Colunas5** | **Colunas6** | **Colunas7** |
| Qtd entrada | Cres. Dist | Cres. Rep | Decres. Dist | Decres. Rep | Alea. Dist | Alea. Rep |
| 120k | 0 | 0 | 4178 | 4245 | 7007 | 2078 |
| 180k | 0 | 1 | 9302 | 9658 | 15738 | 4641 |
| 240k | 1 | 1 | 16628 | 17106 | 8171 | 8216 |
| 300k | 1 | 1 | 26152 | 27682 | 12797 | 12860 |
| 360k | 1 | 1 | 37841 | 39624 | 18556 | 18623 |
| 420k | 1 | 1 | 51326 | 52377 | 25415 | 25352 |
| 480k | 1 | 2 | 67122 | 68659 | 33432 | 33259 |
| 540k | 1 | 1 | 85541 | 87140 | 42500 | 42135 |
| 600k | 2 | 3 | 107122 | 109173 | 52557 | 52294 |
| 660k | 2 | 2 | 131424 | 133504 | 63815 | 63817 |
| 720k | 2 | 2 | 158296 | 161711 | 76837 | 76519 |

**Selection**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Colunas1** | **Colunas2** | **Colunas3** | **Colunas4** | **Colunas5** | **Colunas6** | **Colunas7** |
| Qtd entrada | Cres. Dist | Cres. Rep | Decres. Dist | Decres. Rep | Alea. Dist | Alea. Rep |
| 120k | 5692 | 5686 | 6388 | 14929 | 5868 | 6164 |
| 180k | 12761 | 12780 | 14338 | 33270 | 12880 | 12891 |
| 240k | 22670 | 22697 | 25522 | 59302 | 23238 | 22790 |
| 300k | 35452 | 35458 | 39757 | 92739 | 23238 | 35618 |
| 360k | 51063 | 51079 | 57398 | 133606 | 51373 | 51232 |
| 420k | 69642 | 69672 | 78508 | 182686 | 70309 | 69917 |
| 480k | 91160 | 91215 | 102424 | 237383 | 91783 | 91484 |
| 540k | 116051 | 115328 | 130174 | 302240 | 116202 | 116212 |
| 600k | 143251 | 143977 | 160995 | 372736 | 144559 | 144015 |
| 660k | 174267 | 174089 | 196060 | 450820 | 175284 | 175258 |
| 720k | 208434 | 208099 | 233844 | 537673 | 209211 | 207253 |

**Counting**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Qtd entrada | Cres. Dist | Cres. Rep | Decres. Dist | Decres. Rep | Alea. Dist | Alea. Rep |
| 120k | 0 | 2 | 1 | 2 | 3 | 2 |
| 180k | 1 | 2 | 1 | 2 | 4 | 2 |
| 240k | 2 | 3 | 2 | 3 | 7 | 6 |
| 300k | 1 | 3 | 2 | 3 | 2 | 4 |
| 360k | 2 | 4 | 2 | 4 | 3 | 4 |
| 420k | 3 | 5 | 2 | 5 | 4 | 5 |
| 480k | 3 | 5 | 3 | 5 | 5 | 7 |
| 540k | 3 | 7 | 3 | 6 | 5 | 7 |
| 600k | 3 | 8 | 4 | 6 | 6 | 9 |
| 660k | 4 | 8 | 5 | 8 | 7 | 9 |
| 720k | 4 | 9 | 5 | 8 | 6 | 11 |

**Merge**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Qtd entrada | Cres. Dist | Cres. Rep | Decres. Dist | Decres. Rep | Alea. Dist | Alea. Rep |
| 120k | 13 | 11 | 10 | 11 | 62 | 22 |
| 180k | 18 | 18 | 16 | 139 | 41 | 33 |
| 240k | 25 | 23 | 21 | 22 | 73 | 53 |
| 300k | 144 | 31 | 27 | 30 | 83 | 56 |
| 360k | 53 | 37 | 31 | 35 | 97 | 66 |
| 420k | 62 | 63 | 38 | 41 | 99 | 96 |
| 480k | 53 | 72 | 45 | 46 | 91 | 110 |
| 540k | 57 | 59 | 73 | 54 | 105 | 102 |
| 600k | 64 | 64 | 56 | 60 | 119 | 110 |
| 660k | 70 | 210 | 61 | 91 | 217 | 124 |
| 720k | 75 | 78 | 67 | 70 | 170 | 242 |

**Quick**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Qtd entrada | Cres. Dist | Cres. Rep | Decres. Dist | Decres. Rep | Alea. Dist | Alea. Rep |
| 120k | 12 | 14 | 13 | 14 | 39 | 17 |
| 180k | 18 | 20 | 20 | 18 | 28 | 27 |
| 240k | 24 | 26 | 27 | 25 | 35 | 33 |
| 300k | 39 | 35 | 32 | 31 | 46 | 44 |
| 360k | 44 | 39 | 38 | 40 | 54 | 55 |
| 420k | 45 | 44 | 45 | 46 | 63 | 60 |
| 480k | 50 | 48 | 50 | 50 | 72 | 75 |
| 540k | 53 | 60 | 62 | 58 | 83 | 78 |
| 600k | 63 | 62 | 69 | 65 | 94 | 89 |
| 660k | 66 | 72 | 72 | 71 | 108 | 106 |
| 720k | 73 | 77 | 76 | 80 | 110 | 111 |

**Heap**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Qtd entrada | Cres. Dist | Cres. Rep | Decres. Dist | Decres. Rep | Alea. Dist | Alea. Rep |
| 120k | 11 | 13 | 13 | 12 | 26 | 18 |
| 180k | 19 | 19 | 19 | 19 | 31 | 31 |
| 240k | 26 | 28 | 25 | 26 | 42 | 38 |
| 300k | 33 | 33 | 31 | 31 | 48 | 47 |
| 360k | 41 | 42 | 40 | 40 | 60 | 54 |
| 420k | 49 | 50 | 50 | 48 | 66 | 67 |
| 480k | 56 | 55 | 52 | 53 | 81 | 80 |
| 540k | 59 | 60 | 61 | 59 | 96 | 90 |
| 600k | 66 | 72 | 71 | 68 | 103 | 101 |
| 660k | 73 | 74 | 84 | 77 | 121 | 111 |
| 720k | 81 | 83 | 85 | 101 | 126 | 125 |

**Conclusão dos métodos de ordenação**

**Resumo geral - bubble, insertion e selection**

O bubble, insertion e selection sort são algoritmos de ordenação mais simples porem os mesmos exigem um certo tempo para ordenar, seu tempo escala conforme o volume de entrada.

**Especificidades**

**BubbleSort** trabalha bem com sequência de números quase ordenadas, mas de qualquer maneira conforme o volume de dados o seu tempo vai crescendo exponencialmente.

**InsertionSort** ordena os dados de forma aleatória bem rápido em relação a sua ordenação com números de decrescentes, onde é seu maior gargalo... sequencias de numeros crescentes o tempo é quase zero, fica difícil perceber o seu tempo até em milissegundos.

**SelectionSort** esse algoritmo ordena os números de forma escalar também porém seu gargalo é quando a sequencia é decrescente e repetida conforme o gráfico pode mostrar.

**Resumo geral - Counting, Merge, Quick e HeapSort**

São algoritmos mais complexos, porém a velocidade de ordenação em diversas é indiscutível.

**Especificidades**

**CountingSort** é o algoritmo mais rápido desta analise, porém, ele consome mais recursos dada sua implementação, mas sem dúvida é o menor tempo entre os testes deste trabalho.

**MergeSort** é um algoritmo bem rápido também porém ele tem alguns piques de tempo, talvez seja por causa da máquina ou pelo volume de dados, mas ainda sim é mais rápido do que os 3 primeiros.

**QuickSort** tem um bom desempenho para ordenar numeros em qualquer situação, conforme o gráfico o seu gargalo é quando os números estão distribuídos de forma aleatória.

**HeapSort** segue a mesma condição do QuickSort e praticamente no mesmo tempo.

**Algumas perguntas**

**1 - Qual é o melhor algoritmo quando os elementos estão ordenados de forma  
crescente?**

R:Sem dúvida o InsertionSort, é praticamente imperceptivel o tempo que ele executa o algoritmo na ordem crescente, quase todos resultados estão abaixo de 3 milissegundos.

**2 - Qual é o melhor algoritmo quando os elementos estão ordenados de forma  
decrescente?**

R: CountingSort, esse algoritmo roda em qualquer situação mais rápido em relação aos demais com a base de dados proposta, consequentemente também é o mais rapido ao ordenar de forma decrescente.

**3 - Qual é o algoritmo mais estável em relação ao tempo de processamento? Ou seja,  
qual o que menos varia o tempo de processamento independente da forma como os  
dados estão organizados no vetor?**

R:Quick e HeapSort, independente da entrada, o tempo é quase o mesmo, o que influencia de verdade é a quantidade de entrada de dados.